Claims 1-39 were rejected by the Examiner. New claims 40-60 have been added by the present amendment. Claims 1-7, 9-12, 15-16, and 36-39 have been amended. Claims 1-60 are presently pending. No new matter has been added by the present amendment. The new and amended claims are new, non-obvious, and useful, and are supported throughout the originally filed disclosure. Reconsideration of the present application is requested in light of the amendment, and the remarks given below.

IDS Issues

The Examiner failed to initial all the references on Supplemental IDS which was received by the Patent Office on April 23, 2001 because it was allegedly non-compliant with Patent Office Rules. Five non-patent references were not initialed. The Examiner failed to initial all the references on Supplemental IDS which was received by the Patent Office on January 31, 2002 because it was allegedly non-compliant with Patent Office Rules. The only reference not initialed was German patent 2150506A. The Applicant will submit an appropriate supplemental IDS with these uninitialed references as a separate paper.

The Examiner failed to initial all the references on Supplemental IDS which was received by the Patent Office on Feb. 19. 2002. The Examiner indicated that the IDS was non-compliant with patent office rules. However, the only reference not initialed is the primary reference relied upon in the present office action. Accordingly, Applicant will not re-submit this reference.

Rejection of Claims 35-36 under 35 U.S.C. § 101

Claims 36 and 37 stand rejected under 35 U.S.C. § 101, the Examiner contending they do not recite statutory subject matter. While traversing this rejection, to expedite prosecution, the Applicant has amended these claims to recite an apparatus, which

should moot the rejection.

Withdrawal of the statutory subject matter rejection is respectfully requested.

Rejection of Claims 1-39 under 35 U.S.C. § 102

Claims 1-39 stand rejected under 35 U.S.C. § 103 in light of the Bryant & O'Halleron "15-213 Handout #3:Linking", dated February 13, 2000 (hereinafter "Bryant"). Applicant respectfully traverses the rejection, as discussed below.

The present application discloses a new approach to reordering of software modules that has the potential to greatly streamline the linking process, particularly on systems where the code does not fit entirely in the system memory of the computer system performing the linking operation. In contrast, Bryant is generally directed to a tutorial discussion of the general state of the art of linkers, and neither teaches nor suggests Applicants claimed invention.

The Examiner argues that Bryant anticipates claim 1, in particular Bryant Fig. 14 and page 24.

Applicant's amended claim 1 recites:

> *receiving a software module, the software module including references to locations within the software module, at least some of the references being backward references; and reordering components of the software module to remove at least some of the backward references.*

Claim 1 has been amended to clarify that the backward references being discussed are references found within the software module to elements found or defined in the software module, e.g., data or instruction sections of the software module. Applicant has

also amended the claim to make clear that the reordering involves a rearrangement of components of the software module, e.g., ELF tables, or sections, as recited in some of the newly added dependent claims. Applicant submits that these changes have not altered the scope of the claim 1, as this usage of the terms "references" and "reordering" was clear in light of the original disclosure and original claim language. Applicant has merely made the change to improve clarity and expedite prosecution, not to surrender claimed subject matter.

To anticipate a claim, the reference must teach every element of the claim. *See* MPEP 2131. The identical invention must be shown in as complete detail as is contained in the claim. *See id.* (citing *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236 (Fed. Cir. 1989)). The Examiner posits that the "receiving" element of Applicant's claim 1 is anticipated by Bryant, p. 24, second paragraph, which the Examiner quotes as "For each input file f on the command line, the linker determines if f is an object fie [sic] or an archive backward reference) [sic]". As an initial matter, the Applicant notes that the Examiner has misquoted the cited paper which does <u>not</u> include the words "backward reference" in the position indicated by the examiner. Applicant believes this mistake was probably due to a typographical error and the Examiner meant to indicate that he believed that the archive file was an example backward reference. However, even with the missing parenthetical added, the Applicant respectfully submits that the Examiner's interpretation of the claim language is clearly in error. The archive itself is <u>not</u> a backward reference as recited in Applicant's claim 1, and it is clearly not a reference to a location <u>in the software module</u>, and is also <u>not included</u> in the received software module. Applicant readily admits that the files discussed in Bryant may well contain within them backward internal references. However, these are not backward references <u>included in the received software module</u>, and are not to locations in the received software module. Examiner's argument seems to

be suggesting that the files themselves, or the command line list of files, are the "references" as recited in Applicant's claim 1. Applicant submits that this is inconsistent with the language of claim 1 which clearly indicates that the backwards references at issue are "included in the software module" and are "to locations within the software module". The "f" of Bryant page 24 is a file which contains a software module, software modules, or a part of a software module, not a backward reference included in a software module received for linking.

Second, Applicant's claim 1 recites that the components of the software module are "reordered to remove backward references". Reordering of a software module is neither taught nor suggested by Bryant. The Examiner cites a portion of Bryant which describes symbol resolution with an archive, i.e., the addition of a definition found in an archive to a list of references. Once the correct reference is determined, the symbol is removed from a list of unresolved definitions, and added to a list of resolved definitions. This symbol resolution is not a "reordering of the software module" or a "reordering of the components of the software module". The files cited by the Examiner, the U and D described by Bryant, are temporary stores used by the linker in symbol resolution, not part of a received software module. The file E is merely a list of the object files which are used to build an executable object file. The symbol resolution procedure described by the Examiner is not a reordering of the components of the software module as recited in Applicant's claim 1.

Since several elements of Applicant's claim 1 are missing from the cited reference, Applicant respectfully submits that the claim 1 is not anticipated, and the rejection should be withdrawn.

Claim 2-8 depend, directly or indirectly, from claim 1. Thus, these claims should be patentable for at least the reasons given above for claim 1. Several of the claims have been amended to make them consistent with the language of amended parent claim 1. The "steps of" language has also been removed from several claims as unnecessary surplusage. Claim 5 has also been amended to insert a missing "s", a typographical error.

In addition, Claim 2 recites

>*adjusting at least one of the references in the software module to reflect the reordering of the components of the software module...*

Without giving a detailed identification of where in Bryant the recited "adjusting" is alleged to occur, the Examiner generally indicates that it is "as noted above". But none of the steps described on page 25 of Bryant involve adjusting of the references in a received software module to reflect a reordering of the module or of the components of the module. Rather, the identified change of information in Bryant involves actions with temporary files used by the linker, not the received software module. Applicant respectfully submits that Bryant does not teach or suggest adjusting at least one of the references in a software module to reflect reordering the components of a software module, and thus can not anticipate Applicant's amended claim 2.

In addition, claim 4 recites:

>*wherein the software module includes a symbol table, the software module including no backward references in locations before the symbol table when the reordering the components of software module and adjusting at least one of the references have been completed.*

The Examiner has not specifically identified, and the Applicant has not located, anything in Bryant that teaches or suggest that a software module that has been reordered and adjusted contains no backward references in locations before the symbol table. Applicant respectfully requests the rejection of claim 4 be withdrawn, or at minimum, at least explained with more particularity.

With respect to amended claim 6, amended claim 5 from which claim 6 depends recites receiving the software module in ELF format <u>prior</u> to reordering. Claim 6 recites that the software module remains in ELF format <u>after</u> reordering. Nothing in the cited references or identified by the Examiner teach a reordering of a software module or the components of a software module that

preserves ELF format compatibility. Applicant respectfully requests the rejection of claims 5 and 6 be withdrawn, or at minimum, at least explained with more particularity.

Amended Claim 7 recites:

> *wherein sections in the same segment are*
> *contiguously located in the software module*
> *when the reordering the components of the*
> *software module has been completed.*

Nothing the Examiner identified, or that the Applicant has located, in the cited references teaches or suggests reordering a received software module in a way that results in sections of the same segment being contiguously located after the reordering of the components of a received software module.

Claims 9 is patentable over the cited art for at least reasons similar to those given above for claims 1-8. Moreover, to anticipate a claim, the reference must teach each and every element of the claim. *See* MPEP 2131. The Examiner has not identified, and Applicant has not located, in the Bryant reference a "reorder module" as recited in Applicant's claim 9, or any structure in either Bryant or some other source that is configured to reorder a software module or the components of a received software module in order to remove backward internal references included in the software module.

Applicant also respectfully submits that the Examiner's reliance on inherency is also improper, and traverses the rejection on that ground. No reference is supplied to indicate why Applicant's claim 9 would <u>necessarily</u> be anticipated by the proposed combination. To the extent the Examiner is relying on his own knowledge or "common knowledge", the Applicant traverses the Official Notice and respectfully requests the Examiner cite a reference or provide an affidavit in support of the rejection. See MPEP 2144.03.

Since several elements of Applicant's claim 9 are not found in the cited reference, either expressly or inherently, the Applicant respectfully submits that claim 9 is not anticipated and respectfully requests withdrawal of the rejection of claim 9.

Claims 10-15 depend from claim 9, and thus should be patentable for at least the reasons given above for claim 9.

Claim 16 has been amended to correct a typographical error. Amended Claim 16 recites:

> *receiving a software module sequentially, the software module having at least one symbol reference;*
> *linking the software module onto a target memory space; and*
> *resolving the at least one symbol reference without storing the entire software module in local memory while the symbol reference is resolved.*

The Examiner submits that Bryant section 10 teaches the resolving element of the above claim language. Applicant respectfully disagrees. Nothing in section 10 of Bryant appears to teach or suggest that an entire software module would not be stored in memory for linking and symbol resolution. Before the linker described in Bryant performs symbol resolution, it must necessarily read the software module to be linked. The cited section of Bryant discusses resolving symbol references in a first module that are made to a <u>different</u> module (e.g., a global symbol) without loading the <u>different</u> module into memory. There is nothing to indicate that the entire first module would not have to be stored in memory in order to perform the symbol resolution. Accordingly, Bryant neither teaches nor suggests Applicants claim 16. This special property of the Applicant's claimed invention is made possible because of the particular ordering of the software module to be linked that is obtained by using Applicant's disclosed reordering approach.

Claims 17-22 depend from claim 16, and thus should be patentable for at least all the reasons given above for claim 16.

Applicants claim 23 recites:

> *a linker configured to sequentially receive a software module having at least one symbol reference, the linker configured to resolve the symbol reference, the linker configured to store less than the entire software module in local memory during the resolution of the at least one symbol reference.*

Nothing taught or suggested by Bryant indicates that Bryant's linker would not store an entire software module in memory when the linking of that software module is performed. The very text cited by the Examiner indicates that _entire_ object files, each of which contain one or more software modules, are stored at one time. There is no indication that Bryant contemplates the special form of linking (linking without storing the entire linked software module in memory at one time) that is made possible by Applicant's re-ordering of software modules and that is recited in Applicant's claim 23.

Claims 24 through 35 depend from claim 23 and thus should be patentable for at least the reasons given above for claim 23. Moreover, the Examiner does not cite with particularity how or where Bryant allegedly anticipates these claims. For example, no reference is given to a linker which is configured not to store the data section of a software module in local memory while resolving a symbol reference from the software module, as recited in Applicant's claim 26.

Applicant's claim 36 and 37 generally relate to a specially structured software module which is a result of Applicant's disclosed reordering process. The Examiner does not identify anywhere with specificity how the cited reference teaches or suggests such a software module, or more particularly in claim 36, an ELF software module, with a component located before the symbol table, but with no backward references before the symbol table in the software module. Thus, Applicant respectfully submits that the cited reference does not anticipate Applicant's claim 36 and 37, and respectfully requests withdrawal of the Section 102 rejection.

Claims 38 and 39 have been amended in a similar fashion and for similar reasons as analogous claims 1 and 16. Applicant's claim 38 and 39 should be allowable for at least reasons similar to those discussed above for claims 1 and 16. Applicant respectfully submits that the cited reference does not anticipate Applicant's claim 38 and 39, and respectfully requests withdrawal of the Section 102 rejection.

New Claims 40-60

New claims 40-53 depend from claim 1. They thus should be patentable over the cited references for at least the reasons given above for claim 1. Moreover, Claim 40 recites that software module is reordered <u>prior</u> to linking. This feature is neither disclosed nor suggested by the cited Bryant reference. Claims 47-54 further refine the type of references altered by the claimed method, as well as the effect of the reordering of the software module.
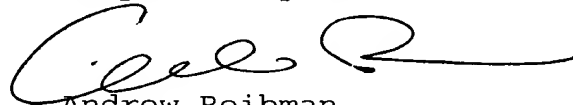
Independent 55 should be allowable over the cited prior art for at least similar reasons to those described above. In particular, the cited Bryant reference does not teach the arranging of the components of a software module so that the order of the components is different than when the software module was received. Claims 56-60 depend from claim 55, and thus should be allowable for at least the same reasons as claim 55.

CONCLUSION

Each of the issues raised by the Examiner has been addressed. It is respectfully submitted that the present application is in condition for allowance. Prompt passage to issuance is requested.

Date: Aug. 13, 2004

Respectfully submitted,

Andrew Reibman
Reg. No. 47,893)

KENYON & KENYON
One Broadway
New York, NY 10004
Phone: (212)-425-7200
Fax: (212)-425-5288

CUSTOMER NO. 26646